Excel for Multi-Unit Auction 2: A Users' Manual (excerpt)*

*The original version of this note was written by Shuya Abe (Graduate School of Information Science, Osaka University) and Naoki Watanabe (Graduate School of Business Administration, Keio University).

Contents

1 Introduction

This material is a user's manual for a software *Excel for Multi-Unit Auction 2* which runs on a computer to implement the Vickrey-Clarke-Groves (VCG) mechanism and run the Greedy-Based Algorithm (GBA) for determining the allocation of some units of a commodity and the payments of bidders in multiple-unit auctions. The GBA is an approximation algorithm developed by Takahashi and Shigeno (2011) which computes the auction outcomes much faster than the VCG.[1]

1.1 Recommended Operating Environment

Excel for Multi-Unit Auction 2 (multi-unit-auction.xlsm) was developed by the authors and is currently (January 2025) available from a github repository. The URL is as follows.

`https://github.com/shuya-abe/excel-for-multi-unit-auction`

In the repository, click on the green button "Code" to open a drop-down list. Then, choose "Download Zip." This is the simplest way to download the file.

The download may be blocked in some corporate or educational environments, because this software uses the macro function of Excel. In such a case, users are asked to download the program via their personal internet environments.

The screen-shots in this manual were taken in the environment indicated in Table 1. *Excel for Multi-Unit Auction 2* runs on computers that can use Excel VBA (Visual Basic for Applications). The recommended environment is listed in Table 2.

Table 1: Operating environment in which screenshots were captured

OS	Windows 10 Enterprise
Excel	Microsoft Office 2016 Excel
CPU	Intel(R) Core(TM) i7-3770K CPU @ 3.50GHz 3.90GHz
Memory	8 GB
Disk	1 TB

[1] Takahashi, S., Shigeno, M., 2011. Approximation algorithms for a winner determination problem of single-item multi-unit auctions. *JSIAM Letters* 3, 29-32.

Table 2: Recommended environment

OS	Windows 7–CMac OS X 10.8–
Excel	Microsoft Office 2013–CMicrosoft Office for Mac 2016
CPU	no requirement if Excel works
Memory	at least 2GB
Disk	no in particular

1.2 Properties of the VCG mechanism

Consider a situation in which multiple units of a homogeneous commodity are auctioned off to bidders. Upon entering the bids for all units into *Excel for Multi-Unit Auction 2* for the settlement of that competitive bid, the program computes an allocation of those units and the payments of bidders. *Excel for Multi-Unit Auction 2* has two computational methods. One is known as the **VCG mechanism** and the other is its approximation algorithm called the Greedy-Based Algorithm (GBA). The computation process in the VCG mechanism is illustrated with an example in the following note.

`http://www.kansai-u.ac.jp/riss/research/publications/public_files/riss_dp/RISS_DP_No68.pdf`

GBA is an approximation algorithm for the VCG mechanism. The explanation is provided in the following note.

`https://www.kansai-u.ac.jp/riss/research/publications/public_files/riss_dp/RISS_DP_No50.pdf`

1.3 Notes

(1) Users need to specify initial conditions for the algorithm, as well as valuations and bids for all bidders for all units, by inputting numerical values in **Arabic Numerals (not double byte characters) without multiple numbers and without missing values**.

(2) *Excel for Multi-Unit Auction 2* uses a program written in Visual Basic. Thus, unlike editing a regular Excel sheet, users cannot restore states by pressing the "backspace" key or by pressing the "Z" key while holding down the "Ctrl" key (`Ctrl-Z`). The Excel file is restored to its state prior to editing by closing the Excel session without saving the file.

(3) The computation results are overwritten each time the program is executed. If you wish to save the results of a computation, then copy them to a different file or make separate copies for each Excel file.

2 Structure of the software package

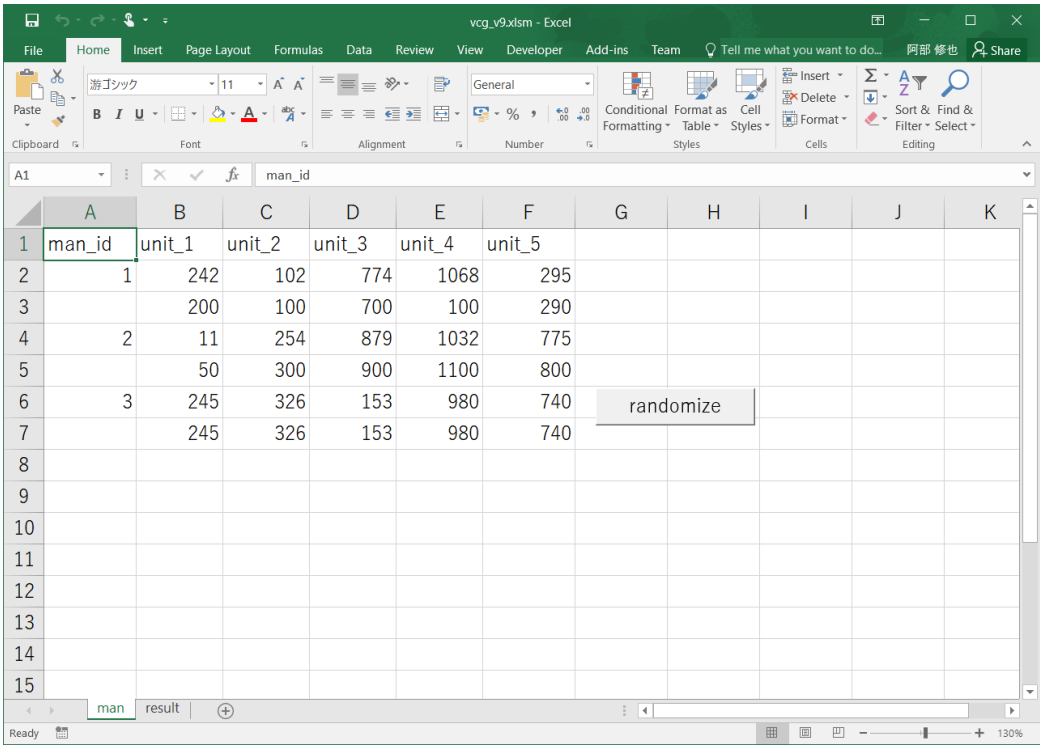
Excel for Multi-Unit Auction 2 consists of two Excel spreadsheets (`man` and `result`) and one VBA program. Here we will describe the content of the two spreadsheets; the usage of the VBA program is briefly explained at the beginning of the next section, while for information on reading and writing VBA code, we ask users to read the standard commercially available reference materials. When running the software, to facilitate computer calculations, we assign each bidder a numerical ID to allow bidders to be easily identified in program input and output.

2.1 man sheet

Each bidder has a certain valuation (the amount that he or she is willing to pay) for each possible subset of the k units available, and each bidder enters bids based on these values. The `man` sheet is used to enter bids in numerical values (not double byte characters) from each bidder for each possible quantity of the commodity. See Figure 1. The entries in this spreadsheet are interpreted as follows.

Column A, `man_id`: For the bidder with ID i (`man_id`=i), his or her ID is entered in row $2i$ of column A in `man` sheet. The IDs are integers that start with 1 and increase consecutively. No bidders have the same ID, and no ID is omitted. The numbers of bidders and units are set by the user when configuring initial conditions (Section 3.1), and the bidder IDs are automatically entered into `man_sheet` accordingly.

Columns B and beyond, `unit_j`: For each bidder in the `man` sheet, enter in each cell the valuation or the bid for each possible unit of the commodity. Based on the settings of the initial conditions (Section 3.1), the available units will be automatically entered in `man` sheet as `unit_j` (`j`=$1, \ldots, k$). For each bidder and each unit, there is one cell for the valuation and one cell for the bid. For the bidder with ID i (`man_id`=i), his or her ID appears in row $2i$ in column A, while valuations are entered in other columns of row $2i$. The value in row $2i$ in the column corresponding to `unit_j` is bidder i's valuation for j units of the commodity. Similarly, the value in row $2i + 1$ of the same column



	A	B	C	D	E	F	G	H	I	J	K
1	man_id	unit_1	unit_2	unit_3	unit_4	unit_5					
2	1	242	102	774	1068	295					
3		200	100	700	100	290					
4	2	11	254	879	1032	775					
5		50	300	900	1100	800					
6	3	245	326	153	980	740	randomize				
7		245	326	153	980	740					
8											
9											
10											
11											
12											
13											
14											
15											

Figure 1: `man` sheet

is bidder i's bid for j units of the commodity. In Figure 1, e.g., for 2 units of the homogeneous commodity (`unit_2`), the bidder with ID 1 (`man_id=1`) has his or her valuation of 120 and enters his or her bid of 100. The program requires that values for both the valuation and the bid be entered for all bidders for all units.

randomize **button** Used to generate random numbers for valuations for the purposes of conducting simulations.

See, in Figure 1, bids for all 3 bidders for all units (from 1 to 3) have been entered into the `man` sheet.

2.2 result sheet

The `result` sheet displays the units allocated to each bidder, the bidder's valuation and payment for units allocated to him or her, and the points he or she earned (valuation minus payment) under the VCG and the GBA, respectively. The various buttons in the spreadsheet are used to execute the calculations in question (Figure 2).

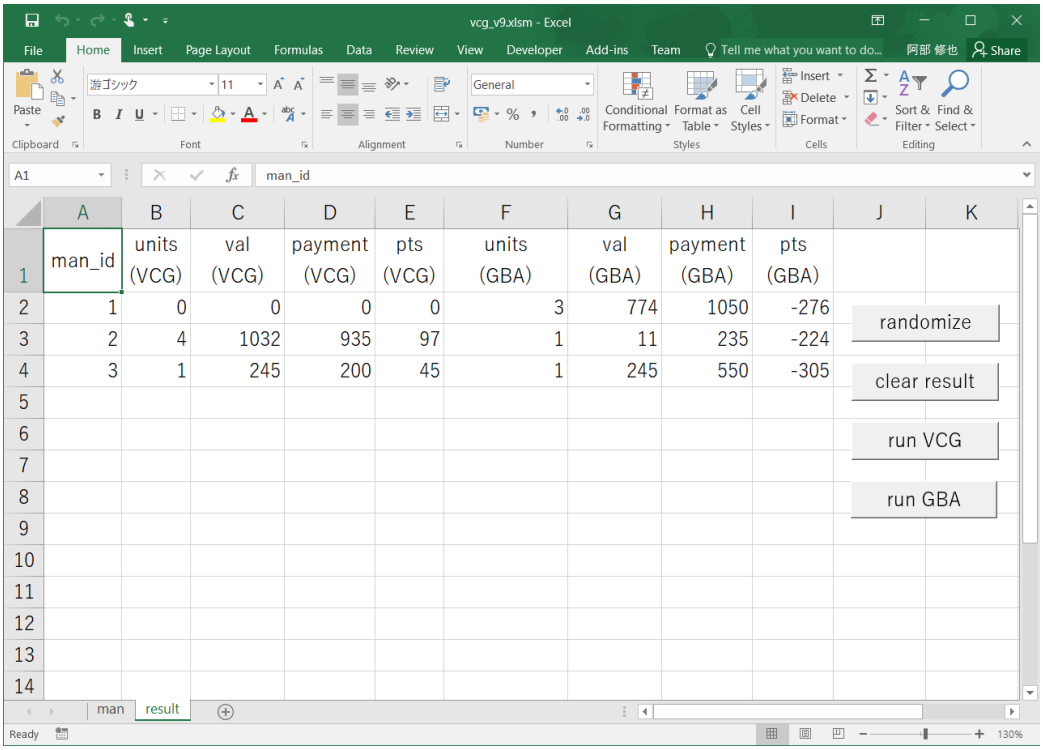


Figure 2: `result` sheet.

Column A: (man_id): Bidder IDs (in ascending order).

Column B (units (VCG)), Column F (units (GBA)): Units allocated to the bidder under the VCG and the GBA, respectively.

Column C (val (VCG)), Column G (val (GBA)): Bidder's valuation for the units allocated to him or her under the VCG and the GBA, respectively.

Column D (payment (VCG)), Column H (payment (GBA)): Payments of the bidder for the units allocated to him or her under the VCG and the GBA, respectively.

Column E (pts (VCG)), Column I (pts (GBA)): Points (the bidder's valuation minus his or her payment) the bidder earned under the VCG and GBA, respectively.

Figure 2. According to the valuations and bids shown in Figure 1, the bidder with ID 2 (man_id=2) was allocated 4 units under the VCG, for which his or her valuation was 1032 and the payment was 935, and thus he or she earned 97 points, whereas under the GBA the bidder was allocated 1 unit, for which his or her valuation was 11 and the payment was 235, and thus he or she earned -224 points. As is shown in the figure, the GBA generates the results which are far from those the VCG generates.

Note that the GBA approximates the results computed by the VCG when all bidders bid their approximately true valuations. The computational procedures of the VCG and GBA are described in the Appendix.

run VCG button: Click on this button to run the calculation of commodity allocation and payments of bidders according to the VCG mechanism.

run GBA button: Click on this button to run the calculation of commodity allocation and payments of bidders according to the GBA.

When the program is invoked to perform calculations, if some valuations are left unspecified in the man sheet, then the calculation may proceed with those valuations set to 0 in some cases. If some bids are left unspecified, then the calculation will terminate prematurely and an error message (Figure 3) will appear. In such a case, click on the **OK** buttonin the error message and fill in values for all missing valuations and bids.

As demonstrated in the Appendix, under the VCG mechanism, the sum of all bids for all bidders and for all units allocated to bidders is maximized; however, there may be more than one such optimal allocation. In such cases,

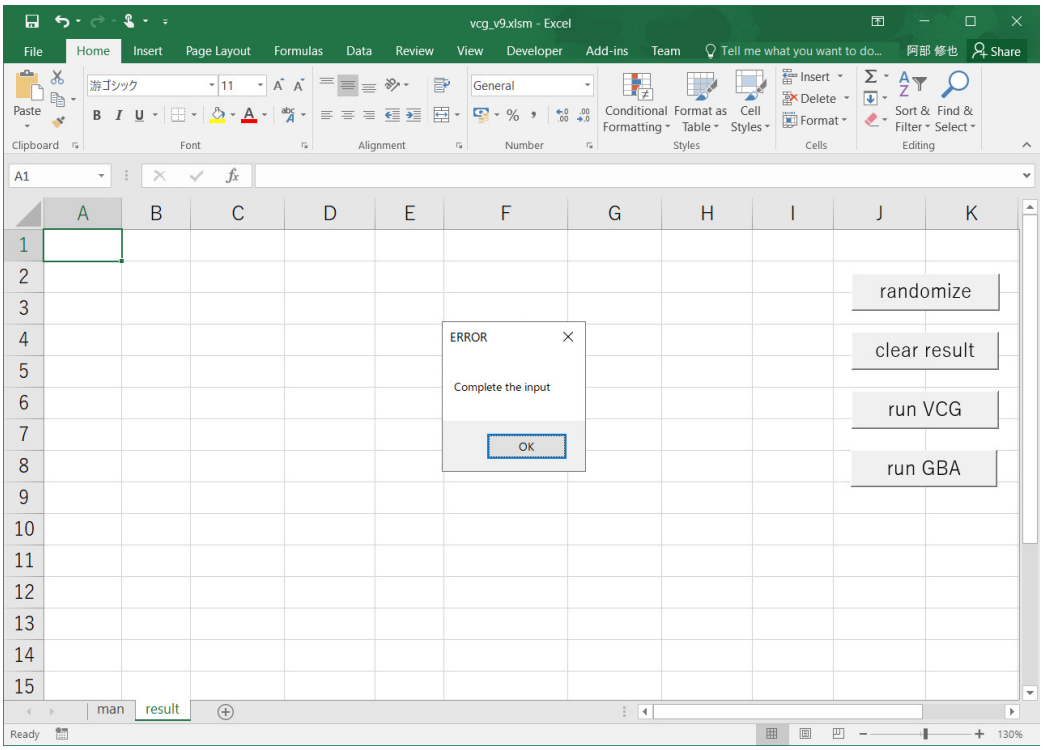


Figure 3: Error message presented in the event of one or more bid values being left unspecified

the program selects an allocation scheme that allocates fewer units to bidders with lower values of IDs. For example, for a case with 3 bidders and 3 units, if there are multiple possible allocation schemes that yield equal values of the total bid, units will be allocated in the order indicated in Table 3.

randomize **button:** Used to generate random numbers for the valuation entries in the `man` sheet (Section 2.1). This button performs the same operation as the randomize button in the `man` sheet.

clear result **button:** Used to clear the results shown on the `result` sheet.

3 Using *Excel for Multi-Unit Auction 2*

3.1 Getting Started

Initially, set the number of bidders n and the number of units k of the homogeneous commodity with the maximum valuation x of the commodity. These values are not set in the spreadsheets but rather within the Visual

Table 3: Order of precedence for cases with multiple equivalent allocations

priority	assigned to bidder 1	assigned to bidder 2	assigned to bidder 3
1	0	0	3
2	0	1	2
3	0	2	1
4	0	3	0
5	1	0	2
6	1	1	1
7	1	2	0
8	2	0	1
9	2	1	0
10	3	0	0

Basic for Applications (VBA) window. In Figure 4, the default values are 3 bidders (`num_man`=3), 5 units (`num_item`=5), and the maximum valuation of the commodity is 300 (`max_value_of_item`=300).

See Figure 5.

1. From the **Development** ribbon, click on the **Visual Basic** button.
2. From **Project Explorer**, open **Module 1** within **Standard Modules** inside **VBAProject (`multi-unit-auction.xlsm`)**.
3. Enter values for Const num_man (number of bidders) and Const num_item (number of units) on the first and second lines, respectively, within **Module 1**. Further, enter the value for Const max_value_of_item (maximum valuation of the commodity) there.
4. **Save** the file and close the Visual Basic window.

3.2 Running the program

1. Set the initial conditions for the number of bidders and the number of units within the VBA window (Section 3.1).
2. In the `man` sheet (Section 2.1), enter valuations and bids for all bidders for all units. In particular, be careful not to leave any bid unspecified.
3. Open the `result` sheet (Section 2.2) and click on the run VCG or run GBA button.

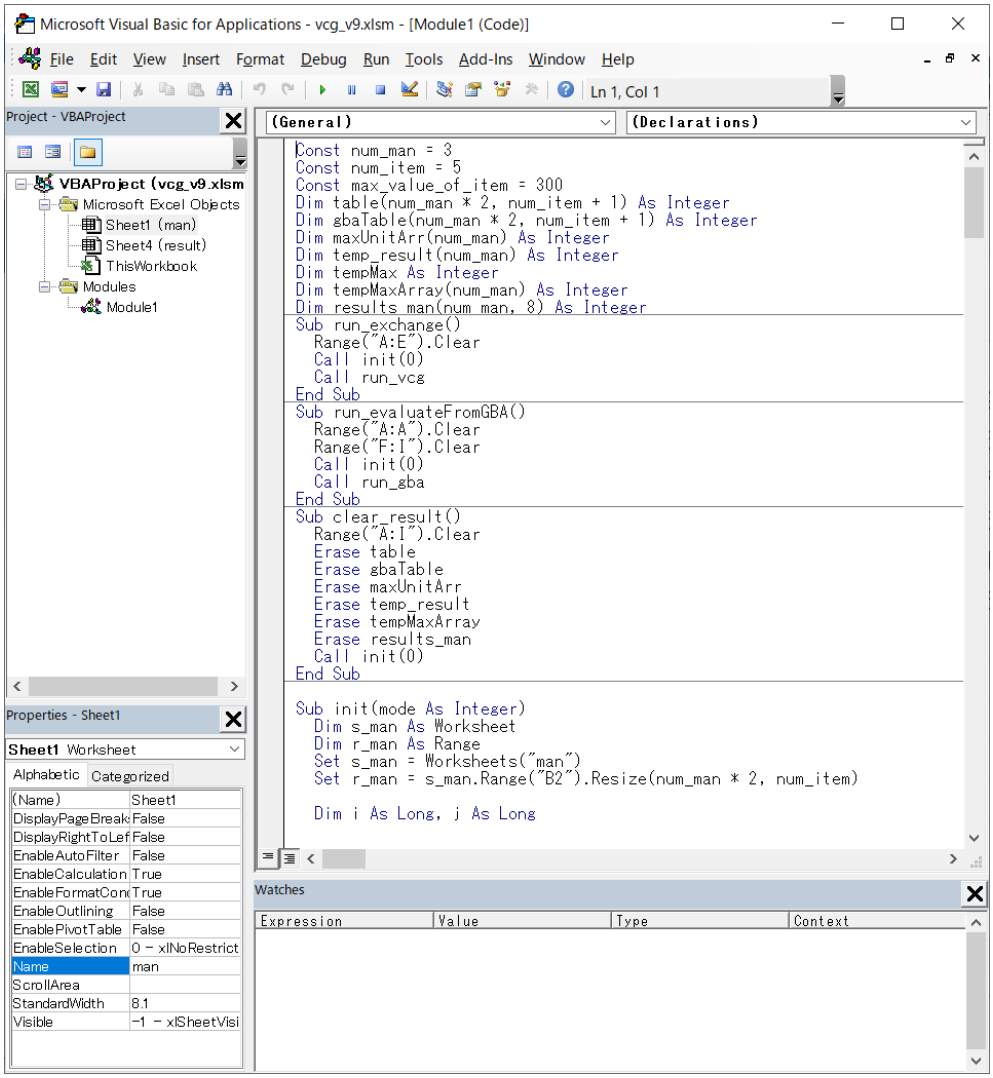


Figure 4: Visual Basic window

3.3 Results

The results computed by the VCG mechanism are shown in Columns A–E within the `result` sheet. In the same sheet, the results computed by the GBA are shown in Columns F–I (Section 2.2). See Section 2.2 for an explanation of the significance of each column.

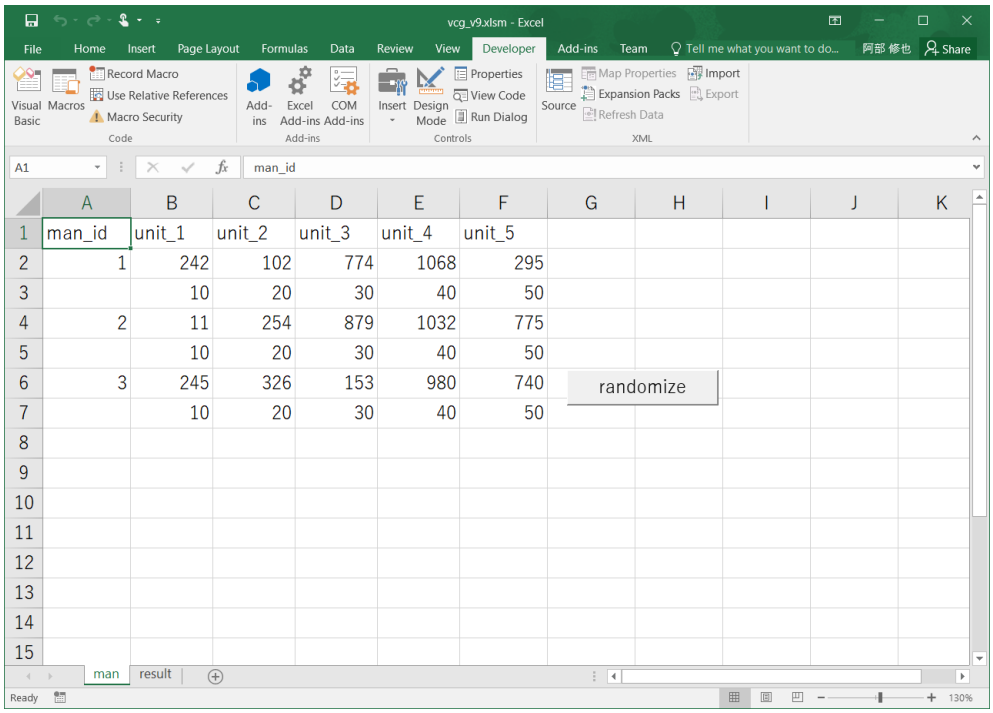


Figure 5: **Visual Basic** button

3.4 Optional: Generating random input data

This software is equipped with the optional capability to use random numbers to conduct simulations. Click on the randomize button in either the `man` sheet (Section 2.1) or the `result` sheet (Section 2.2) to generate random values for the valuations of all bidders for all numbers of units. The numbers of bidders and of units will remain equal to the values specified as initial conditions (Section 3.1).

4 Troubleshooting

In what follows, we discuss remedies for some common problems that may prevent proper execution of the program.

4.1 The Development ribbon is not displayed

The initial settings of Microsoft Office Excel are configured to omit display of the **Development** ribbon. To display this ribbon, proceed as follows (for Microsoft Office Excel 2016).

1. Click on the **File** ribbon.

2. Click on the **Options** button.

3. Within **Excel Options**, click on **Ribbon User Settings.**

4. Confirm that the **Ribbon User Settings** field is set to **Main Tab**, and then check the box for **Development** in the list of ribbons for the **Main Tab** (Figure 6).

5. Click on **OK** and close the **Excel Options window**. The **Development** button should be visible. (If not, close and re-open the Excel file.)

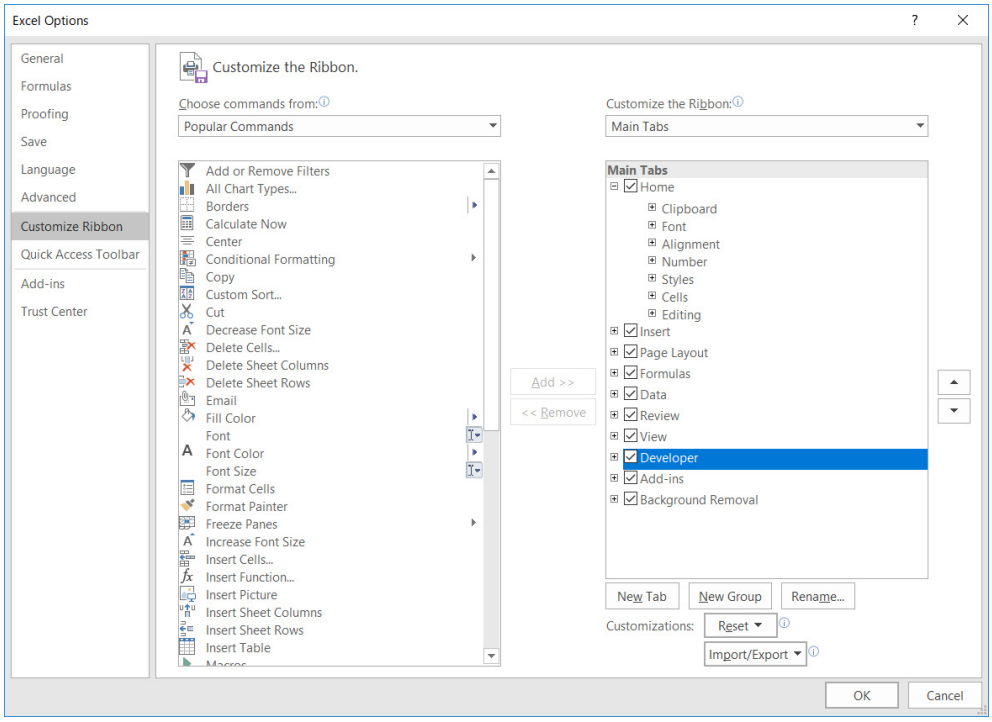


Figure 6: Ribbon user settings

4.2 An error message appears when the program is run

If error messages appear when attempting to run the program, check the following points.

- All numerical input values (numbers of bidders and units in initial conditions (Section 3.1) and valuations and bids in the `man` sheet (Section 2.1) are specified using double byte characters.
- The numbers of bidders and units specified in the initial conditions (Section 3.1) agree with the numbers of bidders and units listed in the `man` sheet (Section 2.1). If the numbers do not agree, clicking on the randomize button will automatically set the numbers to the values specified in the initial conditions.
- There is no missing bid entry.

4.3 No error message appears, but the program cannot compute a correct allocation

Check the following points, if no error message appears but the program does not compute correct outcomes.

- Check whether you opened the program in Protected View (Figure 7). If so, click on **Enable editing (E)** to terminate Protected View.
- Check that the numbers of bidders and units specified in the VBA window (Section 3.1) agree with the numbers of bidders and units listed in the `man` sheet (Section 2.1). If the numbers do not agree, clicking on the randomize button will automatically set the numbers to the values specified in the initial conditions.
- Check that there are no missing valuation entries.

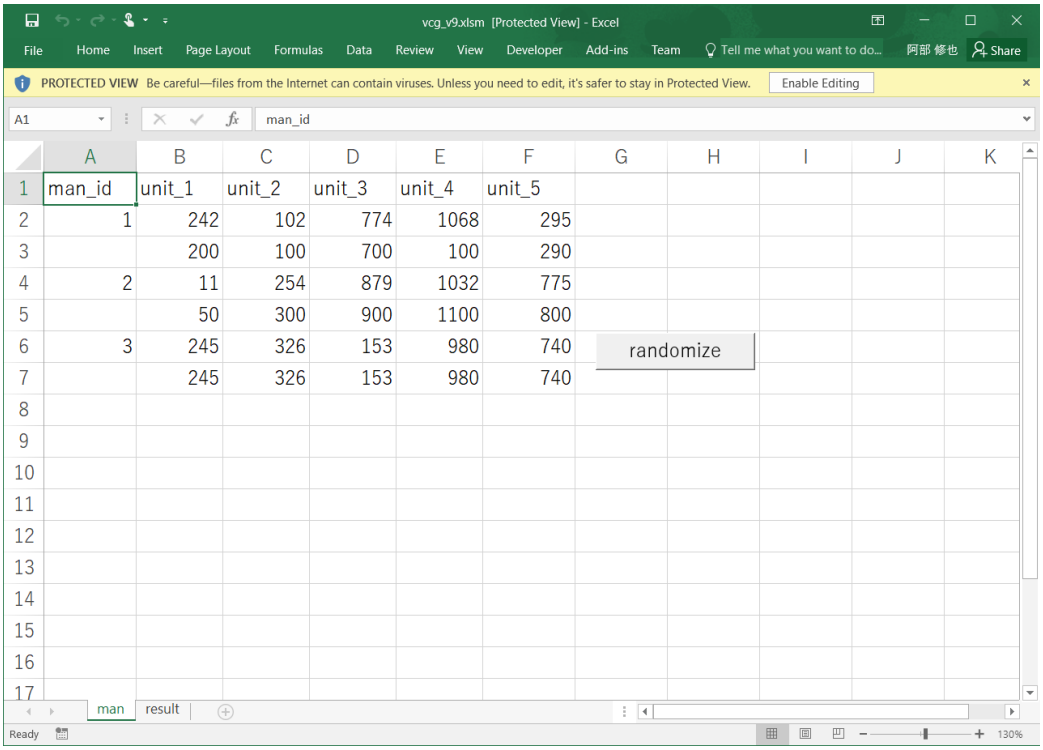


Figure 7: Window as displayed when opened in Protected View